

INFO 6250 SECTION 01  
WENQING LIANG  
001873144  
FINAL PROJEC REPORT

## Simplified Web Music Player

### Project Summary:

In my final project I created a simplified web music player web application. The idea comes from daily music app we use such as Spotify and Pandora, and the functions are being simplified in the project. The application can be used by two types of people: 1) regular users and 2) system manager. The system manager is responsible for managing music database, and regular users can search song database, create playlist, and add the song they like to the playlist.

### Structure:

#### POJOs:

- User [int userid; String username; String password; List<playlist> playlistadded]
- Playlist [int playlistid; String playlistname; List<Song> songinplaylist; User user]
- Artist [int artistid; String artistname; List<Song> songlist]
- Song [int songid; String songname; String url; Artist artist]

DAOs: DAOFactory, MusicDAO, PlaylistDAO, UserDAO

Controllers: PlaylistController, MusicController, UserController, AdminController

AjaxController

### Roles and Functionalities:

System manager: managing music database

- Form-Based Authentication (username & password in Tomcat user file)
- Create and delete artists
- Create and delete songs
- Add songs to an artist
- Review artist lists and song lists
- Play the song (Youtube link)

Regular user:

- Create user account and login (username & password in database)
- Create playlist
- Search database by song name
- Review list of playlists
- Add song to the playlist he likes
- Play the song (Youtube link)

Technologies used:

- Annotations (Controllers, POJOs)
- Entity mapping (one-to-many, many-to-one), corresponding database structure
- Hibernate Query and Criteria (users searching database)
- Form-based authentication
- AJAX (showing “incorrect username/password” for wrong user login)
- URL rewriting, session management

```

package com.music.app.controller;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import com.music.app.dao.DAOFactory;
import com.music.app.dao.UserDAO;
import com.music.app.pojo.User;

@Controller
public class UserController {

    @Autowired
    private DAOFactory factory;

    public UserController(){

    }

    @RequestMapping(value = "/user/login.htm", method = RequestMethod.GET)
    public String showLoginForm() {
        return "user-login";
    }

    @RequestMapping(value = "/user/login.htm", method = RequestMethod.POST)
    public String handleLoginForm(HttpServletRequest request, ModelMap map) {

        UserDAO userdao = factory.createUserDAO();

        HttpSession session = request.getSession(true);

        String username = request.getParameter("username");
        String password = request.getParameter("password");

        System.out.println(username + " " + password);

        try {
            User u = userdao.get(username, password);

            if (u != null) {
                session.setAttribute("currentuser", u);
                return "user-dashboard";
            }
            // else {
            //     map.addAttribute("errorMessage", "Invalid username/password!");
        }
    }

```

```

//            return "user-login";
//        }
    }
    catch (Exception e) {
        System.out.println("Could not login user " + e.getMessage());
    }
    return "user-login";
}

@RequestMapping(value = "/user/logout.htm", method = RequestMethod.GET)
public String logoutUser(HttpServletRequest request) {
    HttpSession session = request.getSession(false);
    if(session!=null){
        session.invalidate();
    }
    return "user-login";
}

@RequestMapping(value = "/user/create.htm", method = RequestMethod.GET)
public String getCreateUserForm(HttpServletRequest request){
    return "user-create";
}

@RequestMapping(value = "/user/create.htm", method = RequestMethod.POST)
public String handelCreateUserForm(HttpServletRequest request){
    HttpSession session = request.getSession(true);

    String username = request.getParameter("newusername");
    String password = request.getParameter("newpassword");

    System.out.println(username + " " + password);

    UserDAO userdao = factory.createUserDAO();
    User newuser = userdao.add(username, password);

    if(newuser!=null){
        return "user-createsuccessful";
    }
    return "user-create";
}

@RequestMapping(value = "/user/delete.htm", method = RequestMethod.GET)
public String deleteUser(HttpServletRequest request) {
    return "user-login";
}
}

```

```

package com.music.app.controller;

import java.util.ArrayList;
import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import com.music.app.dao.DAOFactory;
import com.music.app.dao.MusicDAO;
import com.music.app.pojo.Artist;
import com.music.app.pojo.Song;

@Controller
public class MusicController {

    @Autowired
    private DAOFactory factory;

    public MusicController(){

    }

    @RequestMapping(value="/admin/music.htm", method = RequestMethod.GET)
    public String displayArtists(HttpServletRequest request){
        MusicDAO mdao = factory.createMusicDAO();
        List<Artist> artistlist = mdao.displayArtists();
        HttpSession session = request.getSession();
        session.setAttribute("artistlist", artistlist);
        return "admin-music-managemusic";
    }

    @RequestMapping(value="/admin/music/view.htm", method = RequestMethod.GET)
    public String viewArtist(HttpServletRequest request){

        int artistid = Integer.parseInt(request.getParameter("artistid"));

        MusicDAO mdao = factory.createMusicDAO();
        Artist artist = mdao.getArtistById(artistid);

        HttpSession session = request.getSession();
        session.setAttribute("artistid", artistid);
        session.setAttribute("artist", artist);

        return "admin-music-showartist";
    }

    @RequestMapping(value="/admin/music/add.htm", method = RequestMethod.GET)

```

```

public String addArtistForm(HttpServletRequest request){
    return "admin-music-addartist";
}

@RequestMapping(value="/admin/music/add.htm", method = RequestMethod.POST)
public String handleArtistForm(HttpServletRequest request){

    MusicDAO mdao = factory.createMusicDAO();
    String artistname = request.getParameter("artistname");

    Artist artist = new Artist();
    artist.setArtistname(artistname);
    artist.setSonglist(new ArrayList<Song>());
    mdao.addArtist(artist);

    return "admin-music-addartist";
}

@RequestMapping(value="/admin/music/delete.htm", method = RequestMethod.GET)
public String deleteArtist(HttpServletRequest request){
    int artistid = Integer.parseInt(request.getParameter("artistid"));
    MusicDAO mdao = factory.createMusicDAO();
    mdao.deleteArtistById(artistid);
    return "admin-music-deleteartist";
}

@RequestMapping(value="/admin/music/song/add.htm", method = RequestMethod.GET)
public String addSongForm(HttpServletRequest request){
    int artistid = Integer.parseInt(request.getParameter("artistid"));
    HttpSession session = request.getSession();
    session.setAttribute("artistid", artistid);

    return "admin-music-addsong";
}

@RequestMapping(value="/admin/music/song/add.htm", method = RequestMethod.POST)
public String handleSongForm(HttpServletRequest request){
    HttpSession session = request.getSession();
    int artistid = (Integer)session.getAttribute("artistid");

    String songname = request.getParameter("songname");
    String url = request.getParameter("url");

    MusicDAO mdao = factory.createMusicDAO();
    Artist artist = mdao.getArtistById(artistid);

    Song song = new Song();
    song.setArtist(artist);
    song.setSongname(songname);
    song.setUrl(url);

    mdao.addSong(song);
    return "admin-music-addsong";
}

```

```
@RequestMapping(value="/admin/music/song/delete.htm", method = RequestMethod.GET)
public String deleteSong(HttpServletRequest request){
    int songid = Integer.parseInt(request.getParameter("songid"));
    int artistid = Integer.parseInt(request.getParameter("artistiddelete"));

    HttpSession session = request.getSession();
    session.setAttribute("artistiddelete", artistid);

    MusicDAO mdao = factory.createMusicDAO();
    mdao.deleteSongById(songid);
    return "admin-music-deletesong";
}
}
```

```

package com.music.app.controller;

import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import com.music.app.dao.DAOFactory;
import com.music.app.dao.UserDAO;
import com.music.app.pojo.User;

@Controller
public class AdminController {

    @Autowired
    private DAOFactory factory;

    public AdminController(){

    }

    //display admin error form
    @RequestMapping(value="/error.htm")
    public String showError(HttpServletRequest request){
        return "error";
    }

    //display admin login form
    @RequestMapping(value="/admin/login.htm", method = RequestMethod.GET)
    public String showLogin(HttpServletRequest request){
        return "admin-login";
    }

    //display admin central dashboard
    @RequestMapping(value="/admin/central.htm", method = RequestMethod.GET)
    public String showDashboard(HttpServletRequest request){
        return "admin-dashboard";
    }

    //display all user accounts
    @RequestMapping(value="/admin/user.htm", method = RequestMethod.GET)
    public String manageUserAccount(HttpServletRequest request){
        UserDAO udao = factory.createUserDAO();
        List<User> userlist = udao.displayAll();
        HttpSession session = request.getSession();
        session.setAttribute("userlist", userlist);
        return "admin-user-manageuser";
    }
}

```



```
}

//log out admin
@RequestMapping(value="/admin/logout.htm")
public String logoutAdmin(HttpServletRequest request){
    HttpSession session = request.getSession(false);
    if(session!=null){
        session.invalidate();
    }
    return "user-login";
}
}
```

```

package com.music.app.controller;

import java.util.ArrayList;
import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import com.music.app.dao.DAOFactory;
import com.music.app.dao.MusicDAO;
import com.music.app.dao.PlaylistDAO;
import com.music.app.pojo.Artist;
import com.music.app.pojo.Playlist;
import com.music.app.pojo.Song;
import com.music.app.pojo.User;

@Controller
public class PlaylistController {

    @Autowired
    private DAOFactory factory;

    public PlaylistController(){

    }

    @RequestMapping(value = "/user/playlist.htm", method = RequestMethod.GET)
    public String showAllPlaylist(HttpServletRequest request) {
        HttpSession session = request.getSession(false);
        User u = (User) session.getAttribute("currentuser");
        List<Playlist> list = u.getPlaylistAdded();
        session.setAttribute("currentlist", list);
        return "user-playlist";
    }

    @RequestMapping(value = "/user/playlist/add.htm", method = RequestMethod.GET)
    public String addPlaylistForm(HttpServletRequest request) {
        int userid = Integer.parseInt(request.getParameter("userid"));

        HttpSession session = request.getSession(false);
        User u = factory.createUserDAO().getUserById(userid);
        session.setAttribute("currentuser", u);
        return "user-addplaylist";
    }

    @RequestMapping(value = "/user/playlist/add.htm", method = RequestMethod.POST)
    public String handlePlaylistForm(HttpServletRequest request) {
        PlaylistDAO pdao = factory.createPlaylistDAO();

```

```

HttpSession session = request.getSession(false);

String name = request.getParameter("playlistname");
User u = (User)session.getAttribute("currentuser");

Playlist playlist = new Playlist();
playlist.setPlaylistname(name);
playlist.setSonginlist(new ArrayList<Song>());
playlist.setUser(u);

pdao.addPlaylist(playlist);
//u.getPlaylistAdded().add(playlist);

return "user-addplaylist";
}

@RequestMapping(value = "/user/playlist/view.htm", method = RequestMethod.GET)
public String viewPlaylist(HttpServletRequest request) {

    int playlistid = Integer.parseInt(request.getParameter("playlistid"));
    Playlist playlist = factory.createPlaylistDAO().getPlaylistById(playlistid);
    List<Song> songlist = playlist.getSonginlist();

    HttpSession session = request.getSession(false);
    session.setAttribute("playlist", playlist);
    session.setAttribute("songlist",songlist);

    return "user-viewplaylist";
}

@RequestMapping(value = "/user/playlist/delete.htm", method = RequestMethod.GET)
public String deletePlaylist(HttpServletRequest request) {
    PlaylistDAO pdao = factory.createPlaylistDAO();
    int playlistid = Integer.parseInt(request.getParameter("playlistiddelete"));

    pdao.deletePlaylistById(playlistid);

    HttpSession session = request.getSession(false);

    User u = (User) session.getAttribute("currentuser");
    session.removeAttribute("currentlist");
    session.setAttribute("currentlist", u.getPlaylistAdded());

    return "user-playlist";
}

@RequestMapping(value = "/user/search.htm", method = RequestMethod.POST)
public String searchDataBase(HttpServletRequest request) {
    String keyword = request.getParameter("keyword");

    HttpSession session = request.getSession(false);
    PlaylistDAO pdao = factory.createPlaylistDAO();

```

```

    List<Song> list = pdao.searchSongbyString(keyword);
    session.setAttribute("result",list);
    return "user-searchresult";
}

@RequestMapping(value = "/user/playlist/addsong.htm", method = RequestMethod.GET)
public String addSongToPlaylistForm(HttpServletRequest request) {
    int songid = Integer.parseInt(request.getParameter("songid"));

    HttpSession session = request.getSession(false);
    session.setAttribute("addnewsongid", songid);

    List<Playlist> currplaylists = factory.createPlaylistDAO().displayPlaylists();
    session.setAttribute("currplaylists", currplaylists);

    return "user-addsongtoplaylist";
}

@RequestMapping(value = "/user/playlist/addsong.htm", method = RequestMethod.POST)
public String handleAddSongToPlaylist(HttpServletRequest request) {
    HttpSession session = request.getSession(false);
    int addsongid = (Integer)session.getAttribute("addnewsongid");
    System.out.println("song id: " + addsongid);
    Song song = factory.createMusicDAO().getSongById(addsongid);
    int playlistid = Integer.parseInt(request.getParameter("hiddenid"));
    System.out.println("pl id: " + playlistid);

    PlaylistDAO pdao = factory.createPlaylistDAO();
    Playlist playlist = pdao.getPlaylistById(playlistid);
    playlist.getSonginlist().add(song);
    pdao.updatePlaylist(playlist);
    return "user-addsongsuccessful";
}

@RequestMapping(value = "/user/playlist/deletesong.htm", method = RequestMethod.GET)
public String deleteSongFromPlaylist(HttpServletRequest request) {
    int playlistid = Integer.parseInt(request.getParameter("playlistdelete"));
    int songid = Integer.parseInt(request.getParameter("songiddelete"));

    PlaylistDAO pdao = factory.createPlaylistDAO();
    Playlist playlist = pdao.getPlaylistById(playlistid);

    MusicDAO mdao = factory.createMusicDAO();
    Song song = mdao.getSongById(songid);

    pdao.deleteSongFromPlaylist(playlist, song);
    return "user-viewplaylist";
}

@RequestMapping(value = "/user/back.htm")
public String goBack(HttpServletRequest request){
    return "user-dashboard";
}

```

} }

```

package com.music.app.controller;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;

import com.music.app.dao.DAOFactory;
import com.music.app.dao.UserDAO;
import com.music.app.pojo.User;

@Controller
public class AjaxController {

    @Autowired
    private DAOFactory factory;

    public AjaxController(){

    }

    @RequestMapping (value = "/ajaxservice.htm", method = RequestMethod.POST)
    @ResponseBody
    public String ajaxService(HttpServletRequest request){
        String username=request.getParameter("username");
        String password=request.getParameter("password");

        UserDAO userdao = factory.createUserDAO();
        User u = userdao.get(username, password);

        String result = "";
        if(u==null){
            result = result + "invalid username and password!!";
        }

        return result;
    }
}

```

```

package com.music.app.pojo;

import java.util.List;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.Table;

@Entity
@Table(name="playlisttable")
public class Playlist {
    //property mapping
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @Column(name="PLAYLISTID", unique = true, nullable = false)
    private int playlistid;

    @Column(name="PLAYLISTNAME")
    private String playlistname;

    //entity mapping
    @ManyToOne
    private User user;

    @OneToMany()
    @JoinColumn(name = "playlist_id")
    private List<Song> songinlist;

    public Playlist(){

    }

    public String getPlaylistname() {
        return playlistname;
    }

    public void setPlaylistname(String playlistname) {
        this.playlistname = playlistname;
    }
}

```

```
public int getPlaylistid() {  
    return playlistid;  
}
```

```
public void setPlaylistid(int playlistid) {  
    this.playlistid = playlistid;  
}
```

```
public User getUser() {  
    return user;  
}
```

```
public void setUser(User user) {  
    this.user = user;  
}
```

```
public List<Song> getSonginlist() {  
    return songinlist;  
}
```

```
public void setSonginlist(List<Song> songinlist) {  
    this.songinlist = songinlist;  
}
```

```
}
```



```

package com.music.app.pojo;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.ManyToOne;
import javax.persistence.Table;

@Entity
@Table(name="songtable")
public class Song {
    //property mapping
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @Column (name="SONGID", unique = true, nullable = false)
    private int songid;

    @Column (name="SONGNAME")
    private String songname;

    @Column (name="URL")
    private String url;

    //entity mapping
    @ManyToOne
    private Artist artist; // owning side

    public Song(){

    }

    public Artist getArtist() {
        return artist;
    }

    public void setArtist(Artist artist) {
        this.artist = artist;
    }

    public String getSongname() {

```

```
        return songname;
    }

    public void setSongname(String songname) {
        this.songname = songname;
    }

    public int getSongid() {
        return songid;
    }

    public void setSongid(int songid) {
        this.songid = songid;
    }

    public String getUrl() {
        return url;
    }

    public void setUrl(String url) {
        this.url = url;
    }
}
```

```

package com.music.app.pojo;

import java.util.List;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;

@Entity
@Table(name="artisttable")
public class Artist {
    //property mapping
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @Column(name="ARTISTID", unique=true, nullable=false)
    private int artistid;

    @Column(name="ARTISTNAME")
    private String artistname;

    //entity mapping: non-owning side
    @OneToMany(mappedBy="artist", cascade = CascadeType.ALL, orphanRemoval = true)
    private List<Song> songlist;

    public Artist(){

    }

    public int getArtistid() {
        return artistid;
    }

    public void setArtistid(int artistid) {
        this.artistid = artistid;
    }

    public String getArtistname() {
        return artistname;
    }

    public void setArtistname(String artistname) {
        this.artistname = artistname;
    }

    public List<Song> getSonglist() {
        return songlist;
    }

```

```
}  
  
public void setSonglist(List<Song> songlist) {  
    this.songlist = songlist;  
}  
}
```

```

package com.music.app.pojo;

import java.util.List;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;

@Entity
@Table(name="usertable")
public class User {
    //property mapping
    @Id
    @GeneratedValue (strategy = GenerationType.IDENTITY)
    @Column(name="USERID", unique = true, nullable= false)
    private int userid;

    @Column (name="USERNAME")
    private String username;

    @Column(name="PASSWORD")
    private String password;

    //entity mapping
    @OneToMany(mappedBy="user", cascade = CascadeType.ALL, orphanRemoval = true)
    private List<Playlist> playlistAdded;

    public User(){

    }

    public int getUserid() {
        return userid;
    }

    public void setUserid(int userid) {
        this.userid = userid;
    }

    public String getUsername() {
        return username;
    }
}

```

```
public void setUsername(String username) {  
    this.username = username;  
}  
  
public String getPassword() {  
    return password;  
}  
  
public void setPassword(String password) {  
    this.password = password;  
}  
  
public List<Playlist> getPlaylistAdded() {  
    return playlistAdded;  
}  
  
public void setPlaylistAdded(List<Playlist> playlistAdded) {  
    this.playlistAdded = playlistAdded;  
}  
  
}
```